The Effectiveness of Machine Learning Models on the US Auto Insurance Fraud Detection

Yi Li, Xiyu Chen, Jiawen Yan

***Abstract* — In recent years, the massive volume of the auto insurance industry has resulted in a large number of auto insurance fraud cases. However, traditional manual detection tools are ineffective. This paper compares the predictive performance and the robustness of 4 mainstream machine learning approaches on US auto insurance fraud detection. We divide the US 7-state insurance claim data set into the training, validation, and testing sets. The training set and validation set are used to build the machine learning models, and the testing set is used for model evaluation. Since the number of fraud and non-fraud samples is imbalanced, we resample the training set using SMOTE and train the models over both original and SMOTE training sets. Based on the Cross Validation accuracy and GridSearchCV method, we specify the optimal parameters and combination of hyper-parameters of the learned models and apply them to the same testing set. Comparing the widely-used model evaluation metrics (ROC, AUC, Testing Accuracy, RMSE), we find the Decision Tree Model trained by the SMOTE training set has the strongest prediction power with the highest AUC and Testing Accuracy and lowest errors.**

***Index Terms* — Auto Insurance, Machine Learning, SMOTE Resampling**

I. INTRODUCTION

THE massive volume of the vehicle insurance industry has led to an increasing amount of auto insurance fraud cases in recent years, with insurance firms depending mostly on the judgment of survey experts to identify auto insurance fraud. This strategy is expensive and cannot effectively address the problem of vehicle insurance fraud.

The recent researches on auto insurance fraud are mainly theoretical and empirical, with theoretical researches exploring the causes and possible countermeasures based on game theory. Arrow (1971) introduced the concept of moral hazard to explain fraudulent behavior, and Holmstrom (1979) and Spence and Zeckhauser (1971) studied ex-ante moral hazard and ex-post moral hazard. Mao (2008) found that the information asymmetry will lead to a game relationship between the policyholder and the insurer, and Chen (2014) found that if the number of games is sufficient, policyholders and auto insurance companies tend to obtain long-term benefits through cooperation.

In terms of empirical research, a German insurance company disclosed the claims data from 1994 to 1996. Among those data, 6% of claims were fraudulent claims and 94% were legitimate claims. Phua et al. (2004) proposed that special attention should be paid to the problem of data imbalance when using machine learning methods to identify auto insurance fraud on this dataset. Another data set is the auto insurance claims data of the Massachusetts Automobile Insurance Bureau in 1993. The target variable is an integer variable with a value ranging from integer 0 to 10, which records the degree of suspiciousness of the claim. Brockett et al. (2003) used the PRIDIT method on this dataset to achieve better classification results, but the cost of this method is relatively high.

To sum up, scholars have compared the effects of machine learning methods in identifying auto insurance fraud. These studies demonstrate that machine learning methods have a better recognition effect on auto insurance fraud. However, due to the different claim data sets used to train the model, the optimal model with the best prediction power could be different in these researches. As a result, we must train the well-suited one for our interest data set - US 7-state insurance claim. In addition, since the number of fraud claims samples is far smaller than that of normal claims, we improve this imbalance by using SMOTE resampling on our training set and then train 4 machine learning methods over both the original training set and SMOTE training set.

Our 4 models include Logistic Regression Model (with LASSO), Decision Tree Model, K-Nearest Neighbor Model (KNN), and Support Vector Machine (SVM). Logistic regression is representative of the generalized linear model. The decision tree is representative of the non-distance method. The KNN method is representative of sample learning. The support vector machine model is representative of the nonlinear method. Lastly, we compare the performance of our 4 models using the model evaluation metrics (ROC, AUC, Testing Accuracy, RMSE) and find that the optimal model for our interest data set is the Decision Tree Model trained by the SMOTE training set due to the highest AUC and testing accuracy and lowest RMSE.

The remainder of our paper is organized as follows. The second part is the data preprocessing, especially the SMOTE resampling. The third part is the model construction and analysis, including cross validation and hyper-parameters selection. The fourth part is to evaluate and compare 4 models trained by the original and SMOTE training set over the same testing set. And it also shows our major results and evaluation metrics. The last part is the conclusion and future works.

II. Data Preprocessing

The data set we use is the US 7-state insurance claim data set. In this part, we perform the data preprocessing on the data set, including data description, encoding, splitting, normalization, and SMOTE resampling.

A. Data Description

The data set used in this paper is the 7-state traffic accident claims in the United States in 2015, which was publicly released by Bunty Shah on Kaggle. The data set contains 39 variables (1 target variable and 38 explanatory variables) and 1000 samples. The target variable "fraud_reported" is a binary variable, which records two states of the claims and we encode them into 1 for fraud claims and 0 for normal claims. From the variable description and visualization (Figure 1), normal claims and fraud claims account for 75% and 25% respectively. Among the 38 explanatory variables, " policy_number ", " insured_zip " and " policy_csl " provide limited effective information, so they were deleted. Although the variable "incident_location" records the road name of the place where the traffic accident occurred, it is also deleted because of the widely scattered features, which is not good for prediction. In addition, we also delete three variables " collision_type ", "property_damage", and "police_report_available" that contain a large number of missing values (which are " ? "). The " policy_bind_date " and " incident_date " contain the same information, so we also delete them. Finally, we encode the string data type to the integer data type for model construction.

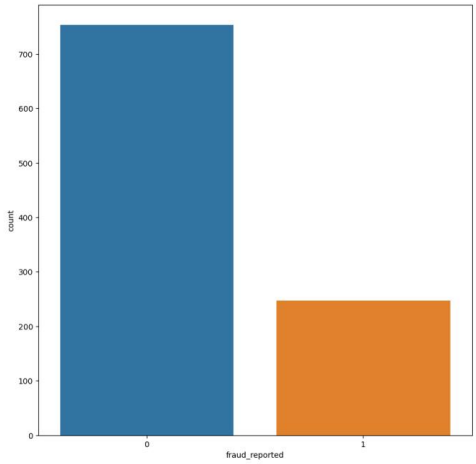

Figure 1: Normal Claims V.S. Fraud Claims

B. Data Splitting

In order to evaluate the performance of the machine learning models, the data set needs to be divided into the training set and testing set: the training set is used to train the machine learning models, and the testing set is used to evaluate the models. After scaling the features, we randomly choose 70% of the samples in the original data set as the training set and 30% as the testing set by sklearn.train_test_split method. Table 1 shows the data splitting result. It can be seen that the proportion of fraud claims in the training set and testing set is very close to the original data set. Thus, both training and testing sets are representative.

TABLE 1
Summary of the Original, Training, and Testing Data Set

	% Fraud Claims	% Normal Claims	Sample Size
Original Set	24.7	75.3	1000
Training Set	24.3	75.7	700
Testing Set	25.7	74.3	300

TABLE 2
Summary of the Original Training and SMOTE Training Data Set

	% Fraud Claims	% Normal Claims	Sample Size
Original Training Set	24.3	75.7	700
SMOTE Training Set	50.0	50.0	1060

C. SMOTE Resampling

As from Table 1, the proportion of fraud claims in the training set is much smaller than that of non-fraud claims. The imbalance between the proportions is not conducive to the model learning and capturing the characteristics, thus affecting the prediction accuracy of the model. Therefore, we use the SMOTE resampling method to resample our training set to make the sample balanced. Table 2 shows the result of SMOTE resampling on the training set. Compared to the original training set, the balance between fraud claims and normal claims after SMOTE resampling is greatly improved.

III. Model Construction

After cleaning, splitting, and SMOTE sampling the training set, we construct the Logistic Regression Model, K-Nearest Neighbor Model, Decision Tree Model, and Support Vector Machine Model over the original training set and SMOTE training set separately, with Cross Validation to adjust the hyper-parameters of each model.

A. Logistic Regression Model

In this part, we utilize the basic logistic regression model and regularized logistic regression model (L1 penalty: LASSO) with 5-fold cross validation over the original training set and the SMOTE training set. Generally speaking, introducing all features into the logistic regression model would lead to the overfitting problem (good performance in the training set and poor performance in the testing set). Hence, we put L1 penalty on the logistic model to partial out the effects of some features with weak predictive power on the label. We obtain 5-fold CV accuracy for each training model, as the following Table 3 and Table 4.

B. K-Nearest Neighbor Model

The KNN method was proposed by Cover and Hart (1967) and it is representative of Lazy Learning. In this part, we identify the choice of K to maximize the cross validation accuracy of the model over the original and SMOTE training set. The small K would lead to the overfitting problem and large variance of the model while large K would cause underfitting issue and large error of the model. Thus, to keep the robustness

of the model, we should select K cautiously. In our model, we randomly split the training set into 70% as a new training set to train the KNN model with different parameters of K (from 5 to 20 with step of 1) and another 30% as the validation set to compare the CV accuracy of different models and select the optimal K with highest accuracy. In the following Figure 2 and Figure 3, we could see that the CV accuracy is highest for the original training set when K = 5, and highest for the SMOTE training set when K = 5 or 6. Thus, we choose the 5-Nearest Neighbor Model for further testing.

TABLE 3
5-FOLD CV ACCURACY (LOGISTIC REGRESSION)

	Original Training Set	SMOTE Training Set
5-Fold CV Accuracy	0.790	0.785

TABLE 4
5-FOLD CV ACCURACY (LOGISTIC LASSO REGRESSION)

	Original Training Set	SMOTE Training Set
5-Fold CV Accuracy	0.797	0.788

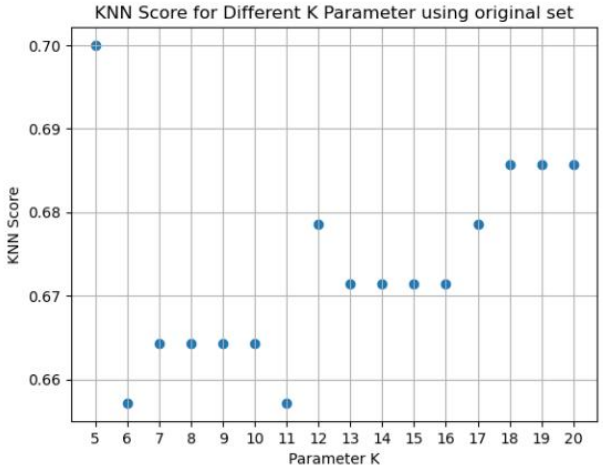

Figure 2: CV Accuracy for the Original Training Set

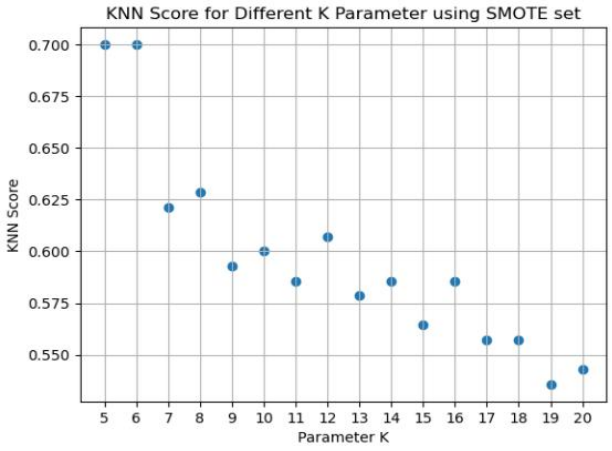

Figure 3: CV Accuracy for the SMOTE Training Set

C. *Decision Tree Model*

The fundamental idea of this model is to split the samples in the training set into different groups based on features. To identify the best decision tree model, we search one combination of the hyperparameters to gain the highest 3-fold CV accuracy using the GridSearchCV function. The optional hyperparameters are: 1. Criterion (the function to measure the quality of a split): "gini", "entropy"; 2. Splitter (the strategy used to choose the split at each node): "best", "random"; 3. Max_depth (the maximum depth of the tree): 5, 10, 15; 4. Min_samples_split (the minimum number of samples required to split an internal node): 2, 3, 4. For the original training set, the best combination of hyperparameters is {"criterion": "entropy", "max_depth": 5, "min_sample_split": 2, "splitter": "best"}. For the SMOTE training set, it's the same. The results of two decision tree models are shown in the following Figure 4 and Figure 5.

D. *Support Vector Machine Model*

This model introduces the concept of kernel to generalize the split of two classes other than only linear lines, making non-linear split possible. It is suitable for the small sample learning method and does not involve probability measure and law of large numbers, simplifying the usual classification and regression problems. Here we use the Kernel Support Vector Machine Model and again use the GridSearchCV function to find the combination of hyper-parameters that gain the highest 5-fold CV accuracy. The optional hyper-parameters are: 1. C (regularization parameter): 10,100,100; 2. Gamma (kernel coefficient): 0.1, 0.01, 0.001; 3. Kernel (specify the kernel type to be used in the algorithm): "rbf", "sigmoid". The results are shown in the following Table 5 with the optimal combination for two training sets and respective 5-fold CV accuracy.

TABLE 5
5-FOLD CV RESULTS OF SVM MODEL

	Original Training Set	SMOTE Training Set
c	100	10
gamma	0.001	0.1
kernel	rbf	rbf
5-Fold CV Accuracy	0.831	0.943

IV. MODEL EVALUATION

In this part, we use 4 widely-used evaluation metrics for each of our models over the testing set: 1. Accuracy: the percentage of correct predictions; 2. Root-Mean-Squared-Error (RMSE): on average how far apart the predicted values are from the ground-truth values; 3. Receiver Operating Characteristic Curve (ROC); 4. Area under Curve (AUC).

A. *ROC Curve*

It is a graph illustrating a classification model's performance with different thresholds and plots the True Positive Rate against the False Positive Rate at corresponding threshold values. If the threshold is low, more samples would be

classified as positive while if the threshold is high, more samples would be classified as negative. The more convex the ROC curve is to the upper left, the better the prediction power of the model. Here are the results of the ROC curves with AUC for each model (Logistic Regression, Logistic Lasso Regression, KNN Model (K = 5), Decision Tree Model, Support Vector Machine Model) that are trained using the original training set and the SMOTE training set separately over the same testing set (Figure 6).

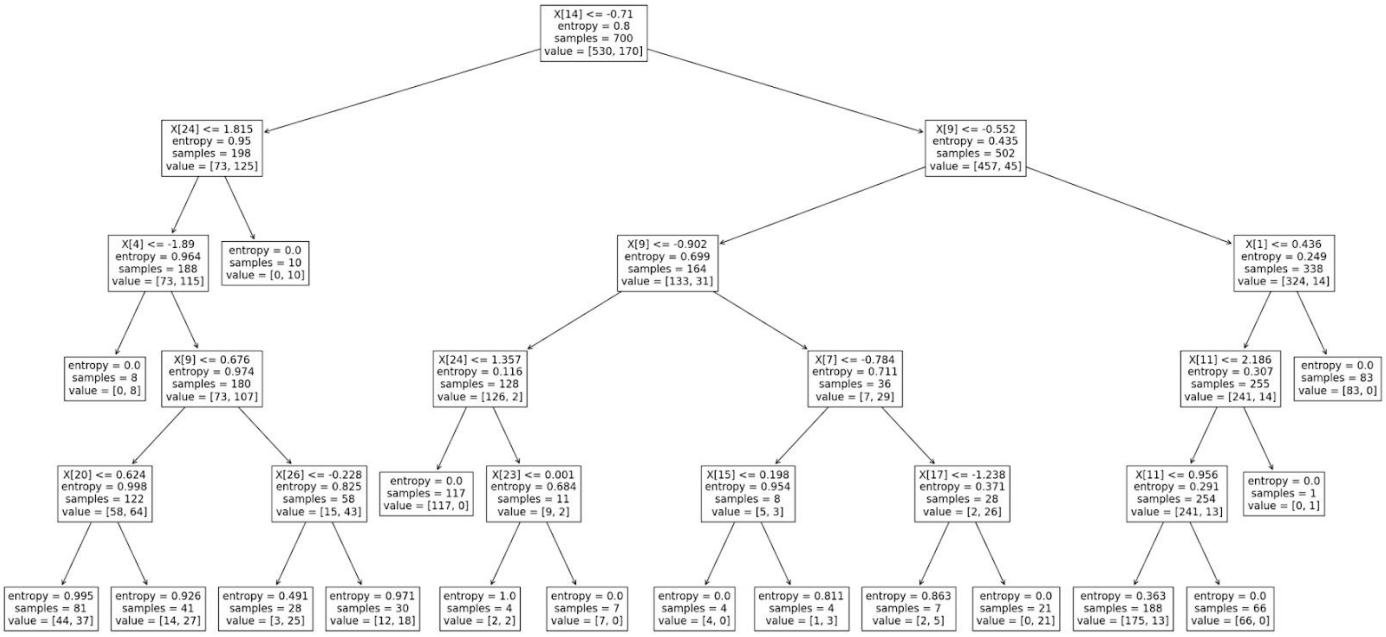

Figure 4: Decision Tree for the Original Training Set

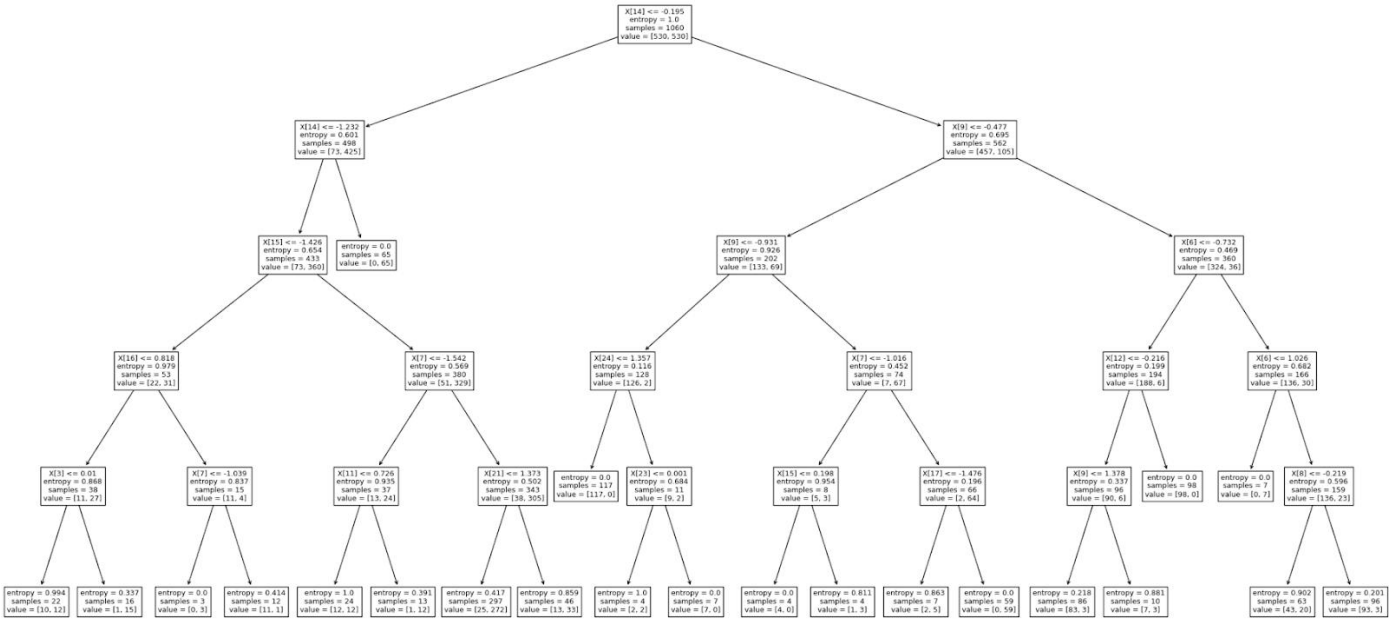

Figure 5: Decision Tree for the SMOTE Training Set

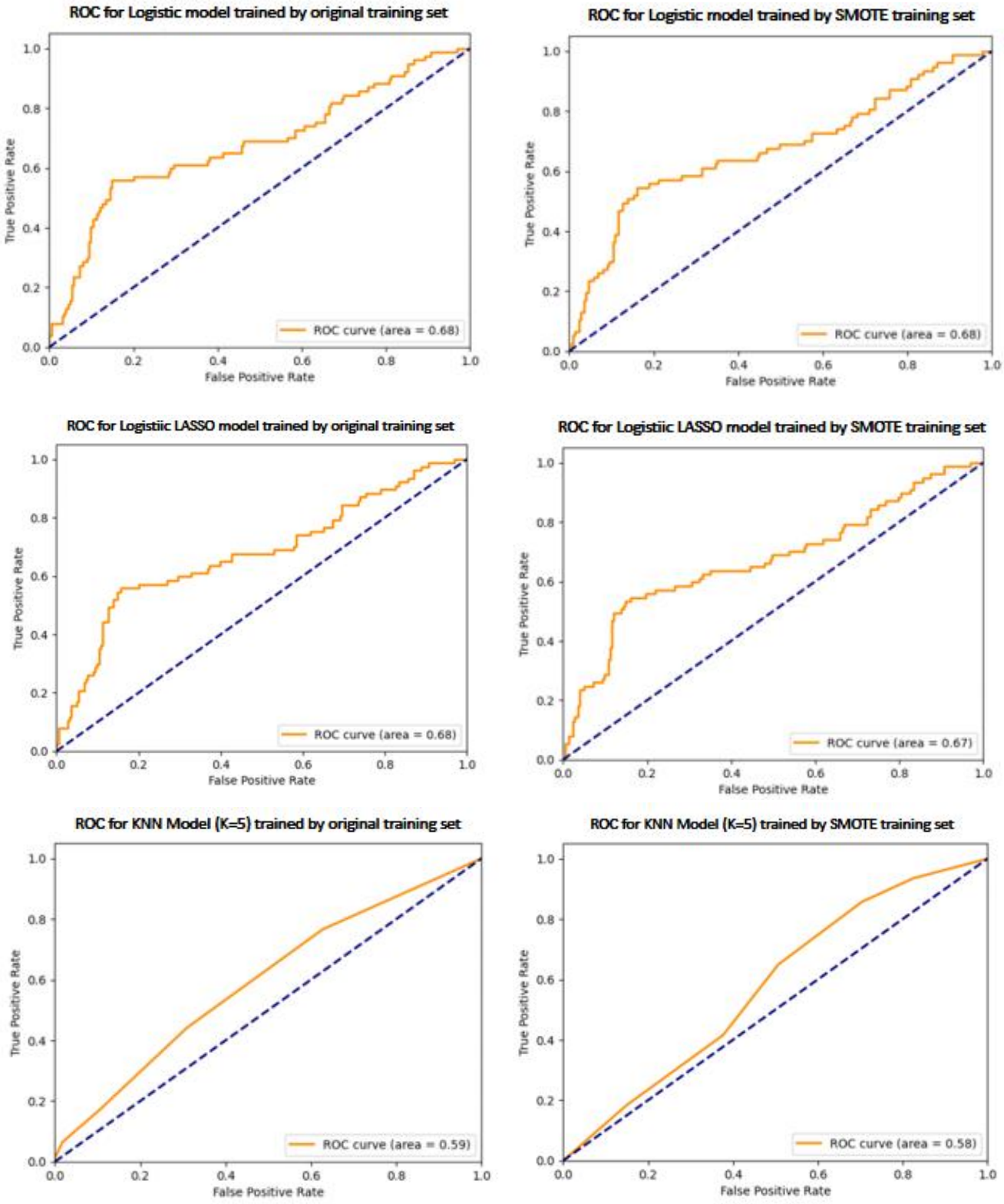

Figure 6: ROC curves with AUC

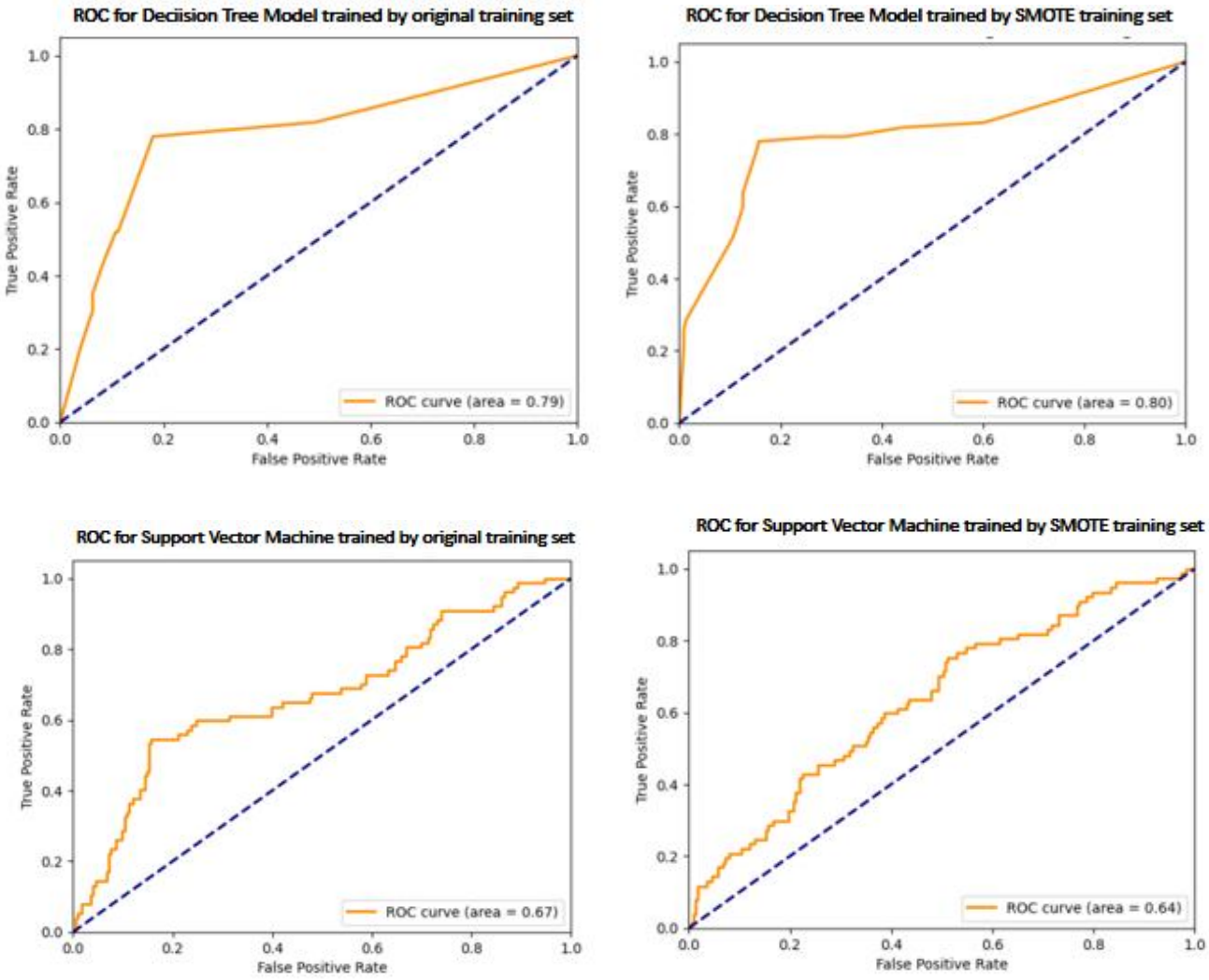

Figure 6 (Cont.): ROC curves with AUC

TABLE 6
EVALUATION METRICS FOR MODELS TRAINED BY THE ORIGINAL TRAINING SET

	AUC			Testing Accuracy			RMSE		
Metrics	AUC	Relative AUC (%)	Ranking	Accuracy	Relative Accuracy (%)	Ranking	RMSE	Relative RMSE (%)	Ranking
Logistic Regression	0.68	86.08	2	0.773	96.99	2	0.476	105.54	2
Logistic Lasso Regression	0.68	86.08	2	0.763	95.73	3	0.486	107.76	3
KNN (K=5)	0.59	74.68	5	0.710	89.08	5	0.539	119.51	5
Decision Tree	0.79	100.00	1	0.797	100.00	1	0.451	100.00	1
SVM	0.67	84.81	4	0.763	95.73	3	0.486	107.76	3

TABLE 7
EVALUATION METRICS FOR MODELS TRAINED BY THE SMOTE TRAINING SET

	AUC			Testing Accuracy			RMSE		
Metrics	AUC	Relative AUC (%)	Ranking	Accuracy	Relative Accuracy (%)	Ranking	RMSE	Relative RMSE (%)	Ranking
Logistic Regression	0.68	85.00	2	0.720	87.48	4	0.529	125.95	4
Logistic Lasso Regression	0.67	83.75	3	0.723	87.85	3	0.526	125.24	3
KNN (K=5)	0.58	72.50	5	0.533	64.76	5	0.683	162.62	5
Decision Tree	0.80	100.00	1	0.823	100.00	1	0.420	100.00	1
SVM	0.64	80.00	4	0.747	90.77	2	0.503	119.76	2

B. AUC, Testing Accuracy, and RMSE

For these three metrics of predictions, we made the following two tables (Table 6 and Table 7) for the four models trained by the original training set and the SMOTE training set over the same testing set. AUC measures how well a model is able to distinguish between classes, and higher AUC implies better predictions. A good model should have higher AUC, higher testing accuracy, and lower RMSE. Regardless of what training set we use to obtain the model to predict, the Decision Tree Model performs the best among the 5 models with the largest AUC (0.79; 0.80), largest testing accuracy (0.797; 0.823), and smallest RMSE (0.451; 0.420). Specifically, the Decision Tree Model trained by the SMOTE training set performs even better with the largest AUC (0.80), largest testing accuracy (0.823), and smallest RMSE (0.420) of all 10 models. In conclusion, the Decision Tree Model trained by the SMOTE training set should be used as a basic model for identifying auto insurance fraud.

V. CONCLUSION

As the development of insurance companies goes fast, the cases of insurance fraud occur more frequently, especially in the market of car insurance, severely jeopardizing the operation of insurance companies and interests of consumers and companies. Recently, there are more challenges emerging in the market of car insurance. Due to inefficient performance of conventional methods to distinguish car insurance fraud, more and more companies started to use machine learning models to distinguish car insurance fraud. In the past researches of car insurance fraud using machine learning models, results are mostly based on one training set and one model, where robustness of the model could not be guaranteed. In this project, we use SMOTE sampling technique to resolve the imbalance issue in the original training set, and build four models (Logistic Regression, Lasso Regression, KNN Model, Decision Tree Model, and Support Vector Machine Model) by using original training set and SMOTE training set separately and comparing their performance. After model construction, we use the ROC curve with AUC, testing accuracy, and RMSE to assess how each model performed compared to each other.

Our project found that the Decision Tree Model trained by the SMOTE resampling training set has the highest AUC, highest accuracy, and lowest RMSE compared to other models. Hence, car insurance companies could consider applying the Decision Tree Model after SMOTE resampling to predict the case of whether it is a fraud. If the company has a very high standard of accuracy of the model, it could use more datasets to test the model and choose the one that performs the best.

Meanwhile, we found other models ranking differently according to the metrics (AUC, accuracy, RMSE), implying that the samples themselves could influence the performance of machine learning models. To perform well, the model should sufficiently use the information coming from the samples, and the Decision Tree Model does it very well in its algorithm.

In terms of other factors that could affect the performance, how well can the feature variables explain the target variable would impact the predictions of car insurance fraud as well. In the future, it could be controlled by analyzing datasets coming from different regions, and that more could be improved by controlling other variables in the datasets. So far, there are projects examining model selection of machine learning to better predict car insurance fraud, but few involve how datasets or samples influence the performance of such machine learning models. In the future, we could focus on variables that could explain well on car insurance fraud, thus improving the model with respect to quality of sample datasets. The application of machine learning models in car insurance fraud detection is in the exploration status, car insurance companies should stay cautious when using the machine learning models especially in the early application period. And insurance companies could not give up the conventional methods right away, and should gradually weigh more on machine learning methods.

REFERENCES

[1] Arrow, K. J.. “Insurance Risk and Resource Allocation. ” Berlin, Springer Netherlands, 1992, 14 : 220 - 229.

[2] Brockett P L, Derrig R A, GOLDEN L L, et al. "Fraud Classification Using Principal Component Analysis of Ridits." Social Science Electronic Publishing, 2003.
[3] Cuixia Chen. "Analysis of Car Insurance Fraud Based on Game Theory in China." Insurance Professional School Paper, 2014, vol.28, no.4, pp. 4.
[4] Holmstrom B. "Moral Hazard and Observability". CORE Discussion Papers RP, 1979, vol 10, no. 1, pp. 74-91.
[5] Phua C, Alahakoon D, Lee V. "Minority Report in Fraud Detection: Classification of Skewed Data." Acm Sigkdd Explorations Newsletter, 2004, vol. 6, no.1, pp. 50-59.
[6] Qin Mao. " Game-theoretic Analysis of Car Insurance Fraud. " Storage Transportation & Preservation of Commodities, 2008, no. 9, PP. 47 - 49.
[7] Spence, Michael, and Richard Zeckhauser. " Insurance, Information, and Individual Action. " Uncertainty in Economics, 1971, vol. 61, no. 2, pp. 380 - 387.